Du Bitcoin aux DAO

Colin, Jean Noël

Published in:
Les blockchains et les smart contracts à l'épreuve du droit

Publication date:
2020

Document Version
le PDF de l'éditeur

Du Bitcoin aux DAO : les fondations techniques de la blockchain

Jean-Noël Colin

Professeur à la Faculté d'Informatique de l'Université de Namur[1]

Introduction

En 2008, Satoshi Nakamoto publiait un article présentant une solution technique pour la mise en œuvre d'une monnaie numérique[2], gérée de manière décentralisée sans nécessité d'un régulateur central et garantissant l'intégrité et la disponibilité du système [9]. Le Bitcoin était né, et avec lui, le concept de blockchain.

Depuis lors, le principe a été repris et développé, pour créer de nouvelles monnaies numériques, mais aussi pour servir de support à des objectifs plus larges, comme les *DAO – decentralized autonomous organization* ou organisation autonome décentralisée dont le fonctionnement est régi de façon entièrement automatique par des programmes informatiques appelés smart contracts (contrats intelligents), qui sont exécutés de manière complètement autonome lorsque les conditions requises sont réunies. La blockchain suscite de l'intérêt et trouve des applications dans de nombreux domaines, qu'il s'agisse de la finance, de la santé, des chaînes d'approvisionnement[3], de la gestion de patrimoine [1,4,11]…

Même si à ses débuts, la blockchain était avant tout un moyen de stocker de l'information de manière sécurisée, transparente et vérifiable, elle est devenue un écosystème applicatif à part entière, permettant de déployer des applications rendues accessibles aux utilisateurs, applications

[1] jean-noel.colin@unamur.be.

[2] On parle de cryptomonnaie pour indiquer le fait que la sécurité du système repose sur des mécanismes cryptographiques.

[3] *Supply chain* en anglais.

s'appuyant sur les capacités d'exécution et de stockage sécurisés offertes par la blockchain.

Dans ce chapitre, nous nous attacherons à décrire les fondements techniques qui sous-tendent une plateforme de type blockchain. Sans entrer dans des aspects avancés de mise en œuvre, nous présenterons les principes de fonctionnement pour permettre de correctement apprécier les enjeux liés à l'utilisation de ces technologies, de juger de leur bien-fondé et d'évaluer les opportunités et leurs risques.

Le reste du chapitre s'organise comme suit : dans la première section, nous décrivons la structure générale d'une blockchain et précisons le vocabulaire utilisé. Ensuite, nous présentons quelques mécanismes cryptographiques fondamentaux pour la sécurité de la blockchain. Les deux sections qui suivent présentent le fonctionnement technique de la blockchain, d'abord comme mécanisme de stockage sécurisé et ensuite comme plateforme d'exécution de programmes informatiques. Nous terminons par une conclusion.

CHAPITRE 1. Description générale d'une blockchain

Le terme « blockchain » est utilisé parfois de façon confuse pour désigner différents éléments de cet écosystème. Il nous parait important de préciser le vocabulaire et de structurer la vision qui prévaudra dans la suite du document.

Nous entendons par blockchain une structure de stockage d'information, sous forme de blocs de données chaînés entre eux. Cette structure de stockage est répliquée sur des *nœuds*[4], formant ce qui est communément appelé un registre décentralisé (*DLT – Distributed Ledger Technology*). Les nœuds sont organisés en un réseau P2P – *peer-to-peer*, dans lequel il n'existe pas d'autorité centrale, mais au contraire, dans lequel le contrôle est exercé conjointement par tous les nœuds.

La gouvernance du réseau est décrite dans la plateforme logicielle (par exemple : Bitcoin, Ethereum, Hyperledger, IOTA…) qui met en œuvre les mécanismes nécessaires. Un élément de cette gouvernance est le protocole de *consensus* qui permet aux participants du réseau de convenir de la validité des informations à stocker sur la blockchain, et ce même en présence de nœuds malicieux ou défaillants.

[4] Il est suffisant pour le propos de décrire un nœud comme un système informatique.

Un nœud remplit différents rôles dans la gestion de la blockchain : il reçoit les transactions des utilisateurs, les valide, les transmet aux autres nœuds du réseau, assemble les transactions validées et stocke les blocs contenant les transactions. Si un nœud complet[5] exerce toutes ces fonctions, d'autres peuvent n'en exercer que certaines ; par exemple, certains nœuds ne servent que de points d'entrée aux transactions, sans participer au processus de validation, certains stockent l'ensemble des données de la chaîne, d'autres ne les stockent pas mais participent à la validation.

L'approche décentralisée a progressivement évolué au cours de la dernière décennie pour permettre non seulement le stockage d'information, mais aussi l'exécution de programmes informatiques, permettant de manipuler le contenu de la blockchain. Ces programmes sont appelés « *smart contracts* », même si dans les faits, il ne s'agit que d'artefacts techniques, n'ayant rien de 'smart' et relevant encore moins du contrat. La prise en charge décentralisée de ces programmes garantit qu'ils seront exécutés complètement jusqu'à leur terme, sans qu'aucune interférence puisse porter atteinte à l'intégrité de leurs résultats. C'est probablement à cette caractéristique que l'on doit le terme de 'contrat', le code du programme définissant les termes de son exécution, la plateforme garantissant le respect intégral de ces termes.

La Figure 1 illustre ces différents concepts en les présentant sous forme de couches reposant sur les services des niveaux inférieurs : à la base se trouve l'infrastructure technique, composée d'ordinateurs interconnectés en un réseau P2P. Sur celle-ci repose le mécanisme de stockage décentralisé. Les informations stockées sur chacun des nœuds sont rigoureusement identiques. C'est le registre décentralisé. Au-dessus se trouvent les services d'exécution décentralisée, qui exploitent et mettent à jour les informations du registre.

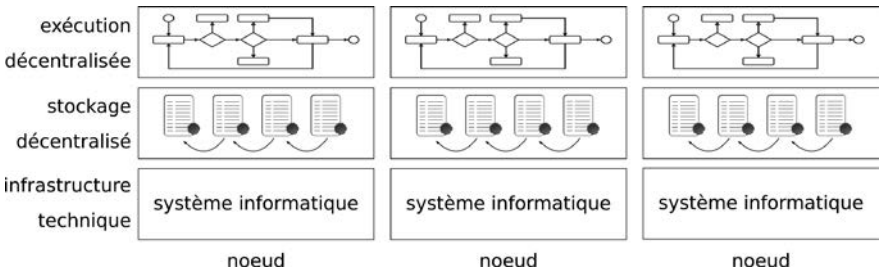

Figure 1 – Modèle de la blockchain

[5] *Full node* en anglais.

Le caractère décentralisé de la blockchain permet de s'affranchir de la dépendance vis-à-vis d'un acteur unique ; le client d'un tel système n'accorde donc plus sa confiance à une entité centrale, mais s'en remet au contraire à la bonne gouvernance du réseau de nœuds pour s'assurer de la véracité de l'information qui est stockée ainsi que de sa disponibilité et sa résilience. En effet, en cas de défaillance d'un des nœuds, il est possible de se tourner vers un autre qui dispose d'exactement la même information et des mêmes capacités.

La blockchain garantit l'intégrité et l'immuabilité des données qui y sont stockées, grâce au recours à des structures de stockage protégées par des moyens cryptographiques. Les informations sont écrites dans des blocs qui sont scellés cryptographiquement, entérinant par là non seulement le contenu du bloc, mais aussi sa relation avec le bloc précédent, figeant de proche en proche le contenu de la chaîne des blocs (d'où le terme « blockchain »).

Ce mode de stockage sécurisé garantit l'immuabilité de ce qui est inscrit sur la blockchain. En effet, modifier un bloc d'information reviendrait à mettre à jour non seulement ce bloc mais aussi tous ses successeurs dans la chaine, ce qui, en fonction du protocole de consensus mis en œuvre, s'avère pratiquement impossible. Cette immuabilité permet une certaine transparence, dans la mesure où il est possible de retracer l'historique de ce qui s'est passé sur la blockchain, garantissant ainsi la vérifiabilité de celle-ci.

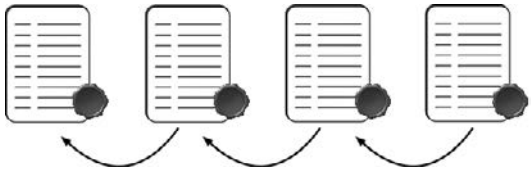

Figure 2 – Chainage des blocs

On distingue les blockchains publiques des blockchains privées[6]. Dans le premier cas, n'importe qui peut se connecter au réseau des nœuds et participer à sa gestion et à son fonctionnement. Il n'y a pas de contrôle d'accès. À l'inverse, le second type de blockchain n'est ouvert qu'à des nœuds connus, authentifiés, relevant d'organisations autorisées.

Selon le type de blockchain, les nœuds peuvent recevoir un incitant à contribuer à la bonne gestion de la plateforme. C'est le cas dans les

[6] *Permissionless* ou *permissioned* en anglais.

blockchains publiques, par exemple Bitcoin ou Ethereum, dans lesquelles les nœuds reçoivent une rétribution financière lorsqu'ils ajoutent un nouveau bloc à la chaîne. Dans une blockchain privée, les participants sont supposés être connus et identifiés, et contribuer au fonctionnement d'un écosystème sans avoir besoin d'un incitant financier direct. Citons à titre d'exemple le cas d'une blockchain rassemblant les acteurs autour d'une chaîne d'approvisionnement, dont l'objet est d'assurer la bonne gestion des données, sans pour autant nécessiter de récompense financière directe pour cette gestion.

Au-delà de la connexion des nœuds, se pose la question de l'accès à l'information stockée sur la blockchain. Différentes approches peuvent exister et se combiner ; si dans une blockchain publique, il parait clair que chacun peut lire ce qui est écrit dans le registre, dans une blockchain privée, il est envisageable de restreindre l'accès à certains participants propriétaires de nœuds, voire à certains utilisateurs finaux, par exemple dans le cas où l'on utiliserait la blockchain pour stocker des informations sensibles auxquelles seuls certains acteurs clairement identifiés pourraient avoir accès.

De plus, l'information stockée dans le registre peut elle-même être protégée, par exemple par des moyens cryptographiques, la rendant lisible uniquement aux personnes possédant une clé de chiffrement valable.

Il est parfois fait référence au caractère anonyme de la blockchain. De nouveau, il s'agit d'une notion à relativiser. Il est vrai que sur Bitcoin, les utilisateurs jouissent d'un anonymat, car ils ne sont désignés que par un identifiant opaque, dérivé de leur clé publique. Aucune autre information personnelle n'est requise pour effectuer des transactions sur cette plateforme. Cet anonymat est cependant relatif, car une transaction pour acquérir des Bitcoins en échange d'une autre monnaie ou la conversion de Bitcoins en une autre monnaie peut laisser des traces diverses, par exemple auprès de l'organisme qui s'est chargé de l'échange. Il est aussi possible de corréler des transactions entre elles sur base de l'identité (anonyme) des leurs participants et ainsi en déduire des informations quant à leur identité réelle. Par ailleurs, rien dans le fonctionnement d'une blockchain n'impose de traiter les utilisateurs de manière anonyme. Il s'agit donc d'un choix de conception et de mise en œuvre.

Finalement, notons que le fait qu'une information soit stockée sur la blockchain ne reflète que le fait que la gouvernance de la plateforme l'a reconnue comme valide, ce qui ne présume en rien de sa véracité.

CHAPITRE 2. Rappels cryptographiques

La sécurité de la blockchain s'appuie sur des primitives cryptographiques qui garantissent la confidentialité, l'intégrité et l'authenticité des informations qui y sont stockées. Nous présentons ici ces mécanismes. Plus d'information peut être trouvé dans [5].

Comme il est de tradition en matière de cryptographie, nous désignons les acteurs par leur nom : Alice et Bob. Nous utilisons les termes « texte », « information » et « donnée » de manière interchangeable.

Le chiffrement est une opération visant à préserver la confidentialité d'une information en la transformant en un texte inintelligible. Pour ce faire, on utilise un algorithme de chiffrement et une clé. Pour retrouver le texte original, on utilise un algorithme de déchiffrement et une clé.

Dans le chiffrement symétrique, Alice et Bob conviennent et au besoin échangent une clé, qui sera utilisée à la fois pour chiffrer et déchiffrer. Alice chiffre le texte à l'aide de cette clé, transmet le texte chiffré à Bob qui le déchiffre à l'aide de la même clé. Le fait que la clé soit partagée pose des difficultés quant à sa génération, sa transmission, son stockage.

Dans le chiffrement asymétrique, chaque acteur est doté de deux clés, l'une publique et l'autre privée. La première peut être diffusée sans contrainte, tandis que la seconde ne peut en aucun cas être divulguée par son propriétaire. L'idée de base est que ce qui est chiffré avec l'une ne peut être déchiffré qu'avec l'autre. Ainsi, pour envoyer un message confidentiel à Bob, Alice chiffre le texte à l'aide de la clé publique de celui-ci, de sorte que seul Bob puisse déchiffrer le texte chiffré grâce à sa clé privée, qu'il est seul à connaître.

Il faut noter que si Alice chiffre le texte à l'aide de sa clé privée, n'importe qui possédant la clé publique d'Alice est en mesure de déchiffrer le texte chiffré. Il n'y a donc aucune confidentialité dans cet échange, mais chacun peut vérifier que le message provient bien d'Alice, autrement dit vérifier son authenticité.

Une autre primitive cryptographique utile à la blockchain est l'empreinte numérique[7]. Une empreinte peut être calculée à partir d'une donnée quelconque, quelle que soit sa taille[8]. L'empreinte a une apparence aléatoire

[7] *Hash* ou *digest* en anglais.

[8] Cela n'est pas tout à fait exact; pour des raisons pratiques, une taille maximale est imposée à la donnée, mais celle-ci est très élevée.

et est compacte, typiquement 20, 32 ou 64 bytes[9], de taille constante, fixée par l'algorithme utilisé[10]. Elle ne révèle donc rien de la donnée originale et possède aussi cette propriété qu'un changement même infime dans la donnée sur laquelle elle est calculée donne lieu à une empreinte radicalement différente. À titre d'exemple, voici l'empreinte de deux phrases pratiquement identiques[11] :

Texte	À l'orée d'une grande forêt vivaient un pauvre bûcheron, sa femme et ses deux enfants.
Empreinte	f565216b85551c349d6208b33e3dbf3eef05c6684a43316ccbdb41346fc37121
Texte	À l'orée d'une grande forêt vivaient un pauvre bucheron, sa femme et ses deux enfants.
Empreinte	aa9625657e4603891a9f866a9580fe330dcd9f59fcb0dbf18efb7f625e1beed4

Même si le nombre d'empreintes possibles est fini, par définition, il n'est pas possible de trouver de collision, soit deux textes qui auraient la même empreinte.

Ce mécanisme permet de vérifier l'intégrité d'une donnée. En effet, il suffit de calculer son empreinte, et de comparer celle-ci avec des empreintes calculées ultérieurement. Toute modification dans l'empreinte indique sans doute possible une altération de la donnée initiale.

En combinant empreinte numérique et chiffrement asymétrique, on obtient un mécanisme garantissant à la fois l'intégrité et l'authenticité d'une information : la signature numérique. Pour assurer ces propriétés au message qu'elle envoie à Bob, Alice calcule l'empreinte du message et chiffre celle-ci avec sa clé privée. Ceci constitue sa signature, qu'elle transmet à Bob avec le texte original. Bob peut vérifier cette signature en la déchiffrant avec la clé publique d'Alice, et en comparant le résultat avec l'empreinte qu'il calcule lui-même sur le texte reçu. Si les deux correspondent, l'intégrité des données est garantie, car si elles avaient été altérées, l'empreinte serait différente. L'authenticité est assurée également, car seule Alice pouvait chiffrer l'empreinte avec sa clé publique. Enfin, Alice ne peut nier avoir signé le texte, ajoutant une propriété de non-répudiation.

L'empreinte et la signature constituent les fondations de la sécurité de la blockchain.

---

[9] Un byte correspond à 8 bits, ou approximativement à un caractère. Un bit est l'unité élémentaire d'information, correspondant à une valeur binaire pouvant prendre la valeur 0 ou 1.

[10] Quelques exemples d'algorithmes de calcul d'empreinte : MD5, SHA-1, SHA-256, RIPEMD160.

[11] La différence entre les deux phrases tient au 'u' de bûcheron, qui est écrit avec et sans accent.

CHAPITRE 3. La blockchain comme mécanisme de stockage

Comme décrit plus haut, la blockchain permet de stocker des informations en s'appuyant sur un réseau de nœuds qui répliquent chacun la même information. La multiplicité de ces nœuds offre des garanties de disponibilité et de résilience : en cas de défaillance d'un, voire de plusieurs nœuds, l'information reste toujours disponible sur le reste du réseau. En l'absence d'une autorité centrale décidant de ce qui peut ou ne peut pas être inscrit dans le stockage, il revient à l'ensemble du réseau de convenir de ce qui constitue une information valide, et donc de l'inscrire dans la blockchain.

SECTION 1. – Structure de la blockchain

À l'inverse d'une base de données classique, dans laquelle il est possible d'ajouter de nouvelles informations, mais aussi de modifier ou supprimer des informations existantes, la blockchain n'offre qu'un moyen de stockage en 'ajout uniquement'. Autrement dit, les informations sur la blockchain sont ajoutées les unes aux autres, en une chaîne formant une séquence *chronologique*. Il n'est pas possible de modifier une information déjà écrite, ceci est garanti par l'usage de fonctions cryptographiques ; toutefois, il est possible de rectifier l'information initiale, mais uniquement en ajoutant une nouvelle version de celle-ci au bout de la chaîne, qui fait référence à l'information à corriger. De cette manière, l'historique des changements est préservé, garantissant la vérifiabilité et la transparence du système.

Les informations inscrites sur la blockchain peuvent être de diverses natures, mais il est commun de les décrire comme des *transactions* : qu'il s'agisse d'un transfert de cryptomonnaie entre deux utilisateurs, de la reconnaissance officielle d'une situation (cadastre de propriété, existence d'un droit intellectuel, occurrence d'un événement) ou d'un accord entre personnes (mariage, échange de bien), tous ces éléments peuvent être repris sous le terme « transaction ».

Les transactions sont regroupées en blocs, de façon similaire aux pages d'un registre papier. Chaque bloc est constitué d'un entête reprenant des informations de gestion (horodatage, taille…) et d'une liste de transactions ; il inclut aussi le hash (voy. Section 3) du bloc précédent. Ainsi, toute modification d'un bloc est interdite, car elle impliquerait le calcul d'un hash différent.

La Figure 3 illustre le mécanisme : le premier bloc (*n*) contient deux transactions ; il contient aussi le hash du bloc précédent (*n*-1). Le hash du bloc *n* est 022656e2. Le bloc suivant (*n*+1) contient aussi deux transactions, et inclut le hash du bloc précédent (*n*). Son hash est f9d37139.

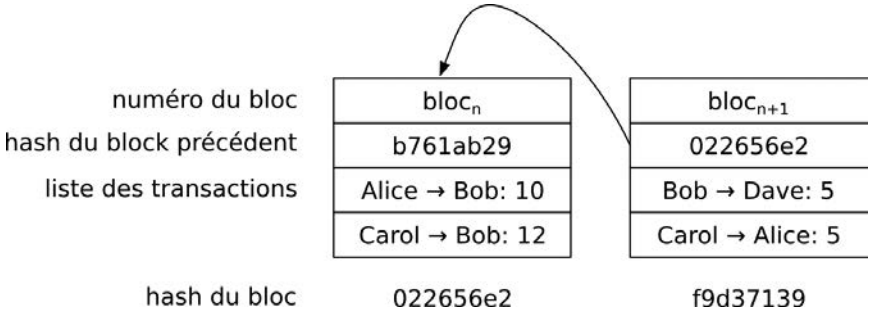

Figure 3 – Chaînage des blocs à l'aide du hash

Comme présenté sur la Figure 4, si l'un des nœuds altère une des transactions du bloc *n*, modifiant par exemple le montant de la seconde transaction, cela donne lieu à un nouveau hash (83ecf615), qui ne correspond plus à celui stocké dans le bloc suivant (022656e2) ; ceci permet la détection de la fraude. Pour restaurer la cohérence de la chaîne, il faudrait recalculer tous les blocs suivants, ce qui est rendu pratiquement impossible par le fonctionnement de la blockchain.

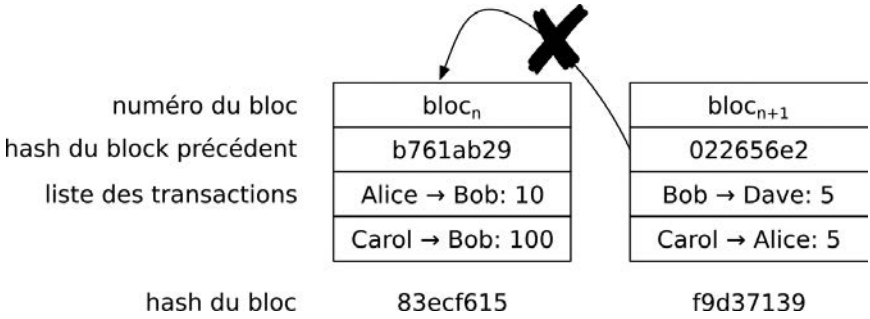

Figure 4 – Incohérence des liens entre blocs suite à une modification d'une transaction

Ce mécanisme de chaînage des blocs garantit donc l'intégrité et l'immuabilité des données stockées sur la blockchain.

SECTION 2. – Ajout de données à la blockchain

Comme indiqué plus haut, les données stockées dans les blocs ne peuvent faire l'objet d'une modification, seul l'ajout de transactions étant autorisé.

Une transaction est soumise à la blockchain par l'un de ses utilisateurs (par exemple un propriétaire de Bitcoin), par l'intermédiaire d'un nœud, qui la transmet aux autres nœuds afin que chacun la valide et l'ajoute à sa propre copie des données.

Cette gestion décentralisée et multipartite soulève différentes questions : d'abord, il convient de valider les transactions soumises, afin de vérifier leur légitimité et leur authenticité. Ensuite, il faut assurer la cohérence des différentes copies de la blockchain maintenue par les nœuds. Enfin, il faut pouvoir garantir le bon fonctionnement du système distribué même en présence de nœuds malicieux ou défaillants.

La validation des transactions est réalisée selon les règles de fonctionnement (de gouvernance) de la plateforme. Ces règles varient d'un usage à l'autre.

Le problème de la cohérence provient du caractère décentralisé de la blockchain. Les nœuds sont interconnectés par un réseau qui offre une vitesse de propagation variable selon la localisation et induit des latences dans les transmissions. Certains liens dans le réseau peuvent aussi être temporairement indisponibles, rendant les nœuds connectés injoignables durant cette période. Ceci impacte la diffusion des transactions et des blocs, et peut provoquer le fait qu'à un moment donné, la liste des transactions à valider, voire même de l'état de la blockchain, diffèrent selon les nœuds.

Le maintien de la cohérence étant un effort collectif, il convient de se prémunir contre le dysfonctionnement de certains nœuds, qu'il s'agisse de panne ou de comportement malicieux, et de permettre la poursuite des activités, même face à des conditions anormales de fonctionnement.

Nous analysons ces différents points dans les sections suivantes.

SECTION 3. – Validation des transactions et constitution des blocs

Lorsqu'une transaction est soumise à un nœud, celui-ci la diffuse aux autres nœuds du réseau. Chaque nœud dispose donc à un moment donné d'une liste de transactions à valider, dites *non-confirmées*. Cette liste peut

varier d'un nœud à l'autre, en raison de la vitesse de propagation des transactions au sein du réseau.

Périodiquement, les transactions sont extraites de la liste, validées et assemblées pour former un nouveau bloc. Le processus de constitution des blocs est appelé *minage*[12], les nœuds qui y contribuent étant appelés les *mineurs*[13]. Chaque mineur extrait de la liste les transactions candidates à l'insertion dans le prochain bloc de la blockchain. Il valide celles-ci en fonction des règles de fonctionnement, et assemble les transactions valides pour former le nouveau bloc. Une fois les transactions inscrites sur la blockchain, elles sont dites confirmées.

La validation des transactions opérées par les nœuds de la blockchain dépend fortement du contexte applicatif. À titre d'exemple, la validation d'un transfert de Bitcoin d'Alice vers Bob consistera à valider que la transaction est authentique, autrement dit qu'elle émane bien d'Alice. Ceci peut être vérifié grâce à la signature numérique (*cf. supra*) qu'Alice appose à la transaction. Les nœuds devront aussi vérifier qu'Alice dispose bien du montant à transférer, ce qui peut se déduire de l'ensemble des transactions passées, stockées sur la blockchain.

Le bloc ainsi constitué est ensuite transmis aux autres nœuds, qui vérifient la validité des transactions qu'il contient et son authenticité, avant de l'ajouter à leur propre copie de la blockchain. Cette vérification est critique, car aucune confiance n'étant postulée entre les nœuds, il est important pour la véracité des informations qu'elles soient validées par chaque participant.

On peut donc résumer le travail d'un mineur à la séquence suivante :

1) Sélectionner les prochaines transactions à inscrire sur la blockchain parmi les transactions non-confirmées
2) Valider les transactions sélectionnées et les assembler dans un bloc
3) Diffuser ce bloc aux autres nœuds du réseau

Cette séquence est répétée à l'infini. Tout en l'exécutant, un mineur est susceptible de recevoir deux types d'information :

- une nouvelle transaction à valider, qu'il ajoute à sa liste de transactions non-confirmées ;
- un nouveau bloc à ajouter à la blockchain ; le mineur valide alors le contenu du bloc, vérifie ses données cryptographiques, et s'assure qu'il s'inscrit bien dans la séquence de blocs, autrement dit qu'il porte bien le numéro suivant le dernier bloc de la chaîne.

[12] *Mining* en anglais.
[13] *Miner* en anglais.

Moyennant ces contrôles, le bloc est ajouté à la copie locale de la chaîne.

Les mineurs travaillent en parallèle et comme décrit plus haut, la liste de transactions non-confirmées peut varier d'un mineur à l'autre. Deux mineurs peuvent donc sélectionner des ensembles différents de transactions non-confirmées, et produire ainsi deux versions différentes du nouveau bloc. Si pendant qu'il construit le bloc *n+1*, un mineur reçoit une version calculée par un autre, il interrompt immédiatement son calcul et accepte le bloc reçu, pour autant qu'il soit valide.

Bien que rare, la situation pourrait se produire où deux mineurs calculent et diffusent quasi simultanément le *n+1*^{er} bloc. Cela conduit à une situation où un nœud dont le dernier bloc est le numéro *n*, recevrait de façon rapprochée deux blocs différents *B* et *B'*, de numéro *n+1*, tous les deux valides et se référant au bloc *n* (voy. Figure 5). On assiste alors à une scission[14] de la chaîne, le bloc *n* ayant deux successeurs possibles.

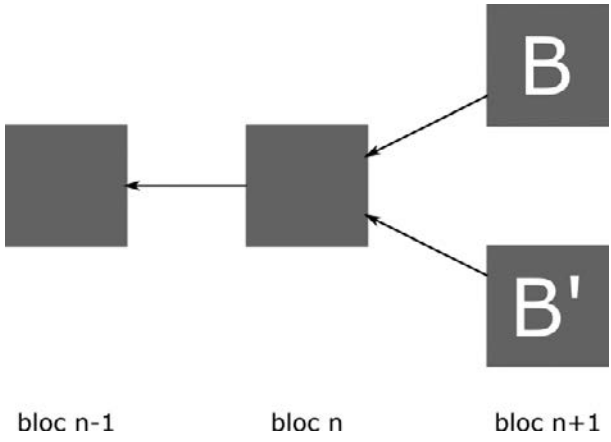

Figure 5 – Scission de la chaîne

Les mineurs de la chaîne calculent le bloc *n+2* sur base de *B* ou *B'*, le premier reçu étant pris en compte. Cela signifie donc qu'au sein du réseau, deux versions de la chaîne se construisent. Dans ce type de situation, un nœud considèrera la chaîne la plus longue comme étant la chaîne valide, ce qui se justifie par le fait que c'est celle dans laquelle aura été investi le plus d'effort de la part des mineurs, la rendant ainsi digne de confiance. Dans la Figure 6, c'est donc la chaîne supérieure qui sera

[14] *Fork* en anglais.

considérée comme valide et c'est sur celle-ci que le nœud basera le calcul du bloc suivant.

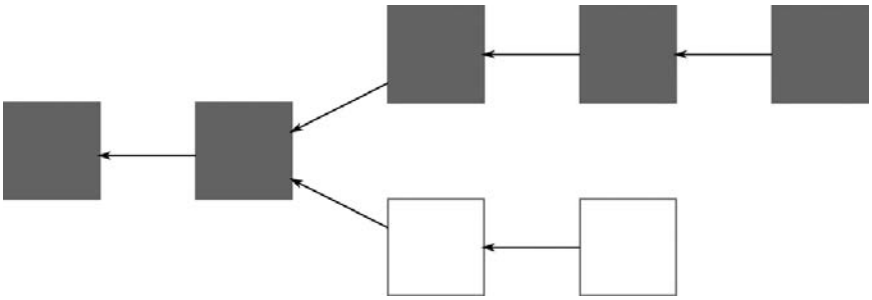

Figure 6 – Validité en cas de scission

SECTION 4. – Consensus au sein de la chaîne

Dans une blockchain publique, et dans une moindre mesure dans une blockchain privée, il n'est pas fait pas d'hypothèse quant à la confiance que les nœuds s'accordent entre eux. Le bon fonctionnement doit être assuré même en présence de nœuds défaillants voire de nœuds malicieux, ce qui fût modélisé dans [8] comme le problème des généraux byzantins, faisant référence à une situation où des généraux encerclant une ville doivent se mettre d'accord sur la stratégie à adopter, en présence possible de traitres et de canaux de communication non sûrs.

Un exemple de comportement malicieux consisterait à exclure des transactions à valider, celles de certains utilisateurs ; un autre serait de soumettre des transactions frauduleuses ; un troisième viserait à modifier la blockchain pour en faire disparaitre certaines transactions ou en inclure de nouvelles.

Le scénario de la double dépense[15] [7] consiste à utiliser plusieurs fois la même cryptomonnaie dans des transactions différentes. On pourrait imaginer qu'Alice soumette deux transactions, l'une transférant l'un de ses Bitcoins à Bob, l'autre transférant le même Bitcoin à un autre compte qu'elle contrôle. Si les deux transactions sont validées par un nœud et diffusées dans le système, les autres nœuds rejetteront ce bloc, car il contient des transactions non valides. Cependant, il est possible que dans

[15] *Double spending* en anglais.

l'intervalle, Bob ait délivré le service ou le bien en échange du Bitcoin qu'il pense avoir reçu, qu'Alice aura ainsi obtenu gratuitement. D'autres scénarios similaires surviennent si Alice soumet les transactions à des nœuds différents qui les intègrent dans des blocs différents. Même si l'une des deux sera rejetée par la blockchain, la contrepartie de la transaction aura peut-être été octroyée.

On le voit, il est possible d'ajouter à la blockchain des blocs contenant des transactions qui n'ont pas été complètement confirmées par l'ensemble des nœuds. C'était déjà le cas lors de scission de la chaîne (voy. section précédente). Il est donc prudent d'attendre qu'une transaction se trouve dans un bloc enfoui à une certaine profondeur[16] dans la blockchain avant de la considérer comme définitivement validée. Par exemple, sur la chaîne Bitcoin, la profondeur recommandée est de 6.

Faute de confiance dans les autres nœuds, la blockchain doit implémenter un mécanisme garantissant que même dans des conditions adverses, les nœuds honnêtes de la chaîne s'accorderont sur le contenu de celle-ci. C'est l'objet de l'algorithme de *consensus*.

L'algorithme le plus connu est celui de la *preuve de travail*, ou *Proof of Work*. C'est celui utilisé dans Bitcoin ou Ethereum. Dans cette méthode, pour prouver sa bonne foi, le nœud qui propose un nouveau bloc doit démontrer qu'il y a consacré un effort suffisant, l'effort étant mesuré en ressources de calcul consommées par le nœud. Un nœud malhonnête sera donc dissuadé de soumettre un bloc frauduleux vu le coût important de l'opération.

Concrètement, chaque nœud doit solutionner un problème complexe à résoudre (le *travail*) mais dont la solution peut être rapidement vérifiée. Cela permet aux mineurs qui reçoivent le nouveau bloc avec la solution du problème lié de rapidement vérifier la validité de celle-ci avant de l'incorporer à leur chaîne.

La capacité à résoudre le problème est principalement une question de chance ; dans le cas du Bitcoin, il s'agit de trouver un nombre (appelé *nonce*) tel que lorsque l'on calcule le hash du bloc combiné à cette valeur, l'empreinte obtenue présente un préfixe de 0 d'une longueur choisie, qui représente la difficulté du problème. Intuitivement, le lecteur réalisera aisément que chercher un nombre tel que lorsque combiné au bloc, leur empreinte commence par un seul 0 est plus facile que si l'empreinte doit commencer par dix 0. Il faudra dans ce dernier cas essayer avec beaucoup plus de nonces avant de tomber sur un qui satisfera au critère.

[16] La profondeur d'un bloc est définie par le nombre de blocs qui le suivent dans la blockchain.

Si tous les mineurs disposaient de la même puissance de calcul[17], ils auraient une probabilité identique de résoudre la preuve de travail. Ce n'est cependant pas le cas ; qui plus est, certains mineurs se coalisent en *pools* de mineurs, augmentant ainsi leurs chances d'identifier le nonce correct.

Il faut noter que si plus de la moitié des mineurs sont malicieux, en se concertant, ils disposent d'une puissance de calcul leur permettant d'imposer leur propre version de la blockchain aux autres mineurs. C'est ce que l'on appelle l'attaque de la majorité ou « des 51 % »[9].

Pour gérer l'évolution de la puissance de minage à l'œuvre dans la blockchain, la difficulté du problème à résoudre est ajustable. Ainsi, si plus de mineurs sont actifs, la difficulté sera augmentée ; à l'inverse, si la puissance totale de minage diminue, la difficulté sera réduite, l'objectif étant de maintenir constante la vitesse de production des blocs[18].

Le principal reproche adressé à la preuve de travail est l'énergie considérable utilisée dans le processus [7]. Chaque mineur effectue en effet des opérations de calcul lourdes, et le travail d'un seul mineur se voit couronné de succès, les ressources de calcul consommées par les autres l'ayant été en vain.

D'autres algorithmes de consensus ont donc vu le jour, dont nous décrivons les principaux rapidement ci-dessous.

Dans la *preuve d'enjeu* (*Proof of Stake*), le nœud souhaitant participer au minage doit faire la preuve de son intérêt par le dépôt bloqué par le système d'un montant (de cryptomonnaie) en garantie. En échange, il devient mineur et obtient le droit de participer à une loterie désignant le mineur chargé du calcul du prochain bloc. La probabilité d'être désigné est proportionnelle au montant de la garantie déposée.

Une approche similaire est la *preuve d'espace* (*Proof of Space*) dans laquelle pour participer au minage, un nœud doit faire la preuve qu'il dispose d'un certain volume de stockage.

Dans la *preuve de temps écoulé* (*Proof of Elasped Time*), chaque participant définit un délai aléatoire durant lequel il suspend ses activités. À l'expiration du délai, il reprend le processus de création de bloc.

Dans un consensus par *preuve d'autorité* (*Proof of Authority*), la création des blocs est la prérogative de participants identifiés et reconnus

[17] On parle aussi de puissance de hashage ou *hashpower* car l'opération réalisée est le calcul d'un hash.

[18] Par exemple, dans Bitcoin, un nouveau bloc est produit en moyenne toutes les dix minutes. Si la puissance des mineurs est telle que ce délai se raccourcit, on augmente automatiquement la difficulté du problème pour conserver ce délai.

auxquels l'ensemble des nœuds délègue le processus de validation. Ce modèle s'éloigne donc d'une approche sans confiance (*trustless*), puisque la construction de la blockchain est la prérogative d'un sous-ensemble de nœuds. L'intérêt de cette approche est son efficacité. Elle ne convient pas pour une blockchain publique, mais peut être utilisée dans une blockchain privée dans laquelle les acteurs se connaissent.

Dans les méthodes qui précèdent, le hasard joue un grand rôle ; un élément aléatoire est en effet pris en compte pour désigner le mineur qui minera le prochain bloc.

D'autres approches plus formelles sont aussi mises en œuvre dans certains systèmes, qui se basent sur un protocole de vote durant lequel les participants se mettent explicitement d'accord sur le contenu du prochain bloc. Un exemple d'une telle méthode est l'algorithme *Practical Byzantine Fault Tolerant* [3]. Dans ce type d'algorithme, on ne repose plus sur une sélection aléatoire des mineurs, mais sur un protocole de négociation entre les nœuds, qui élisent le nouveau bloc. Le problème de ce type d'approche est qu'il est difficile applicable à de larges réseaux, et est donc plus adapté pour des blockchains privées.

CHAPITRE 4. La blockchain comme mécanisme d'exécution

Dans la section suivante, nous avons présenté la blockchain comme étant un mécanisme permettant de stocker des informations, en garantissant leur intégrité, leur disponibilité, leur immuabilité et leur vérifiabilité.

Cette approche a ensuite naturellement évolué vers des services d'exécution de code (ou programme) sur la blockchain, garantissant que lorsque les conditions sont remplies, un programme est exécuté par la blockchain, de façon garantie, soit sans que quiconque puisse interférer avec sa bonne exécution ni l'interrompre. C'est le concept de smart contract, parfois traduit par 'engagement auto-exécutant'.

Le terme smart contract est attribué à Nick Szabo [12]. Dans cet article, il faisait le postulat que « de nombreuses clauses contractuelles peuvent être inscrites dans le logiciel et le matériel que nous utilisons quotidiennement, de manière à rendre coûteux tout manquement au contrat ». D'après l'auteur, la conception d'un smart contract doit répondre à quatre objectifs :

– la possibilité pour les parties d'observer la manière dont les autres exécutent leur part du contrat, et de leur prouver la bonne réalisation de leurs propres obligations ;

– des éléments probants concernant la bonne exécution du contrat ou au contraire relatif à une brèche dans celui-ci doivent être disponible pour un arbitre tiers ;
– l'information concernant le contrat lui-même et son exécution doit être accessible uniquement aux parties qui y ont un intérêt légitime ;
– le caractère exécutoire du contrat doit être garanti.

Cette conception visionnaire trouve dans la blockchain une plateforme capable de rencontrer ces différents objectifs. Bien que la blockchain Bitcoin intègre un mécanisme programmable permettant d'associer à une transaction des conditions de validité (on parle de *script* Bitcoin), le langage de programmation est trop rudimentaire pour parler de smart contract. C'est la chaîne Ethereum qui a introduit un langage de programmation dit « Turing complet », suffisamment puissant et général pour exprimer quelque comportement que ce soit à exécuter sur la blockchain.

Un contrat intelligent est donc un programme informatique, qui est stocké sur la blockchain ; le contrat est écrit dans un langage de programmation, par exemple Solidity ou Viper sur Ethereum, JavaScript ou GoLang pour Hyperledger.

La blockchain garantit qu'une fois déclenchée, l'exécution du contrat se poursuivra jusqu'à son terme, suivant le code écrit, sans interférence. Les résultats produits par le contrat sont stockés sur la chaîne sous forme de transactions, de façon à rencontrer les objectifs de transparence et de preuve. De plus, le contrat étant conservé sur la blockchain, il peut être lu et vérifié par les acteurs autorisés (voire par tous dans le cas d'une chaîne publique). Le site « Ethereum Directory »[19] reprend une liste de contrats définis sur sa blockchain.

Un exemple simple de contrat serait celui d'un prêt entre Alice et Bob ; Alice transfère à Bob un montant, et Bob s'engage à restituer ce montant au terme du prêt et à payer un intérêt tous les mois. Dès l'enregistrement et l'activation du contrat sur la blockchain, celle-ci se charge de manière autonome de son exécution. En fonction des paramètres choisis pour la durée du prêt et le taux d'intérêt, toutes les transactions sous-jacentes sont générées de manière autonome et automatique par la blockchain, garantissant ainsi la bonne exécution du contrat.

Cet exemple simple n'utilise que des informations disponibles sur la blockchain ; dans d'autres cas, le contrat aura besoin d'avoir accès à une information externe à la chaîne. Prenons le cas d'un contrat d'indemnisation de voyageur en cas de retard de vol ou d'un pari sportif. L'information de retard ou le résultat de la compétition ne provient pas de

[19] https://etherscan.io/directory.

la blockchain ; le contrat doit donc l'obtenir d'une source externe, mais néanmoins digne de confiance. Cette source est appelée *oracle*. Il s'agit d'une entité extérieure à la blockchain, mais offrant des garanties quant à la véracité et la vérifiabilité de l'information fournie. Des organisations comme *Provable*[20] ou *TLSNotary*[21] fournissent de tels services.

Un smart contract peut nécessiter la manipulation et le stockage de données trop volumineuses pour être conservées sur la blockchain (rappelons que les données stockées sur la blockchain font l'objet de l'établissement d'un consensus selon un processus coûteux). Ces informations sont alors maintenues dans une base de données, appelée *world state*, elle aussi répliquée par les nœuds, et la blockchain ne conserve que l'empreinte cryptographique (voy. Rappels cryptographiques) des modifications à ces données.

Dans une blockchain publique, l'exécution du contrat peut requérir une contrepartie financière ; celle-ci compense le coût d'utilisation des ressources d'exécution du contrat et de stockage de ses données. La plateforme Ethereum utilise le mécanisme de *gaz* pour mesurer et rétribuer la quantité de ressources consommées par l'exécution d'un contrat.

Un smart contract peut donc être créé pour répondre à des besoins de gestion variés : qu'il s'agisse d'une nouvelle cryptomonnaie sur une chaîne existante, de données d'état civil ou de cadastre de propriétés, de paris ou de contrats d'assurance, voire même de desseins criminels comme décrit dans [6]. Tous ces processus reposent sur le fonctionnement décentralisé de la blockchain.

On désigne par le terme *Application Décentralisée* (*DApp – Decentralized Application*) l'assemblage des services offerts par les contrats et l'interface utilisateur permettant d'y accéder. Cette interface est indépendante de la blockchain mais interagit avec elle au travers d'invocation de services déclenchant l'exécution des contrats intelligents. Le site Etherscan[22] intègre un répertoire d'applications de ce type. On y retrouve des applications aussi diverses que l'élevage et l'échange d'animaux virtuels (CryptoKitties), un service d'annuaire (Ethereum Name Service) ou une place de marché pour biens numériques (OpenSea).

À une échelle plus large, le concept de *DApp* peut être étendu à la gestion d'une organisation entière, ce que l'on appelle une *Organisation Autonome Décentralisée* ou *DAO – Decentralized Autonomous Organization*. Ce terme décrit une organisation dont le fonctionnement est intégralement régi

[20] https://provable.xyz/.
[21] https://tlsnotary.org/.
[22] https://etherscan.io/dapp.

par des DApps et des contrats intelligents. Le résultat est une organisation dont l'intégrité des processus est garantie, de même que leur transparence. De plus, ces processus sont exécutés de façon complètement automatique et autonome.

La cryptomonnaie Dash est gérée par une DAO du même nom[23] ; Augur[24] est une DAO offrant des services de plateforme de marché ; la DAO Digix[25] propose des services de trading d'or en échange de leur propre monnaie (DGX).

Un autre exemple d'organisation autonome décentralisée est TheDAO ; cette organisation, implantée sur la blockchain Ethereum, offrait des services de *crowdfunding* proposant à ses membres de voter pour des propositions de projets à financer, et une fois le vote clos, d'exécuter les transactions nécessaires selon la décision de la majorité des membres. Cette organisation s'est rendue célèbre pour l'attaque qu'elle a subie [2], qui, en exploitant un bug dans un contrat, a mené au vol de l'équivalent de près de 60 millions USD, ainsi que pour les débats qui ont suivi quant à la poursuite des opérations sur la blockchain, certains considérant que le code est la loi (« *Code is Law* ») et que le vol étant la conséquence de l'exécution légitime d'un contrat erroné, il n'y avait aucune raison d'annuler ce contrat, d'autres demandant la rectification de la chaîne pour en effacer les transactions frauduleuses. Ce débat a conduit à une scission de Ethereum en deux chaînes concurrentes.

Les différents exemples présentés dans cette section démontrent l'usage de la blockchain comme plateforme sécurisée, permettant d'exécuter de manière transparente, vérifiable, autonome, des processus qui lui sont soumis. Les termes *smart contract*, *DApp* ou *DAO* couvrent des réalités identiques, mais à des échelles différentes.

Conclusion

Dans ce chapitre, nous avons présenté la blockchain comme un écosystème technologique offrant des services sécurisés, vérifiables, reposant sur un réseau de gestion décentralisé. Ces caractéristiques ont ouvert la porte à des modèles économiques et de gestion nouveaux, dans lesquels

[23] https://www.dash.org/.
[24] https://www.augur.net/.
[25] https://digix.global/.

l'affranchissement vis-à-vis d'une autorité centrale au profit d'une confiance dans un réseau de pairs est un aspect saillant.

Depuis la création de la première blockchain en 2008, les services ont évolué, du stockage de données jusqu'à l'exécution de programmes et au support complet de processus organisationnels, permettant le fonctionnement autonome et automatique d'une organisation entière. Un grand nombre de plateformes ont vu le jour, avec des bonheurs différents.

Nombreux sont les acteurs dans des domaines aussi variés que la santé, l'économie, l'administration… à s'interroger sur l'intérêt que pourraient avoir de telles technologies pour leur fonctionnement et à initier des projets. Il convient de mener cette réflexion en profondeur ; ces nouveaux modèles ne sont pas applicables à tous les contextes, la maîtrise de la complexité de ces environnements demande des compétences avancées, et l'adoption des technologies n'est pas anodine, en termes de coûts et de ressources nécessaires. De plus, le domaine est en continuelle évolution, activé par une intense recherche scientifique, ce qui incite à la prudence au moment de se lancer dans de nouveaux projets.

Bibliographie

[1] G. Bello et A. J. Perez, 2019, « Adapting Financial Technology Standards to Blockchain Platforms », in *Proceedings of the 2019 ACM Southeast Conference* (ACM SE '19), Association for Computing Machinery, Kennesaw, GA, USA, 109-116. DOI:https://doi.org/10.1145/3299815.3314434.

[2] M. del Castillo, 2016, « The DAO attacked: Code issue leads to $60 million ether », theft *Coindesk,* June 18.

[3] M. Castro et B. Liskov, 1999, « Practical Byzantine Fault Tolerance », in *Proceedings of the Third Symposium on Operating Systems Design and Implementation* (OSDI '99), USENIX Association, USA, 173-186.

[4] W. Chen, Z. Xu, S. Shi, Y. Zhao and J. Zhao, 2018, « A Survey of Blockchain Applications in Different Domains », in *Proceedings of the 2018 International Conference on Blockchain Technology and Application* (ICBTA 2018), Association for Computing Machinery, Xian, China, 17-21. DOI:https://doi.org/10.1145/3301403.3301407.

[5] J.-N. Colin, 2016, « Du secret à la confiance… quelques éléments de cryptographie », in *L'identification électronique et les services de confiance depuis le règlement eIDAS*, Larcier, 7-28.

[6] A. Juels, A. Kosba et E. Shi, 2016, « The Ring of Gyges: Investigating the Future of Criminal Smart Contracts », in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (CCS '16), ACM, New York, NY, USA, 283-295. DOI:https://doi.org/10.1145/2976749.2978362.

[7] G. O. Karame, E. Androulaki et S. Capkun, 2012, « Double-Spending Fast Payments in Bitcoin », in *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (CCS '12), Association for Computing Machinery, New York, NY, USA, 906-917. DOI:https://doi.org/10.1145/2382196.2382292.

[8] L. Lamport, R. Shostak et M. Pease, 1982, « The Byzantine Generals Problem », in *ACM Transactions on Programming Languages and Systems* (July 1982), 382-401.

[9] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system,* http://bitcoin.org/bitcoin.pdf.

[10] K. J. O'Dwyer et D. Malone, 2014, « Bitcoin mining and its energy footprint », in *25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)*, 280-285. DOI:https://doi.org/10.1049/cp.2014.0699.

[11] S. Saberi, M. Kouhizadeh, J. Sarkis et L. Shen, 2019, « Blockchain technology and its relationships to sustainable supply chain management », *International Journal of Production Research* 57, 7 (April 2019), 2117-2135. DOI:https://doi.org/10.1080/00207543.2018.1533261.

[12] N. Szabo, 1996, « Smart Contracts: Building Blocks for Digital Free Markets », *Extropy* 8, 1 (1996), 50-53.

[13] R. Zhang, R. Xue et L. Liu, 2019, « Security and Privacy on Blockchain », *ACM Comput. Surv.* 52, 3 (July 2019). DOI:https://doi.org/10.1145/3316481.